Enron Emails Classification

Introduction

Enron Corporation was a Houston-based energy company that was once one of the largest companies in the United States. Founded in 1985, Enron was known for its innovative business practices and was at the forefront of the deregulation of the energy industry in the 1990s. The company traded in natural gas, electricity, and other commodities, and also provided energy services and products to customers around the world.

However, in 2001, the company collapsed in one of the biggest corporate scandals in American history. It was revealed that Enron had engaged in widespread accounting fraud, including manipulating financial statements to hide debt and losses, and overstating its profits. The scandal led to the conviction of several executives, including CEO Jeffrey Skilling and chairman Kenneth Lay, on charges of securities fraud and other crimes. The company filed for bankruptcy in December 2001, and its assets were sold off in the following years to pay off its creditors.

The Enron emails dataset is a collection of emails and attachments from Enron Corporation. The dataset includes over 500,000 emails from approximately 150 Enron employees, including senior executives, that were collected as part of a federal investigation into the company's financial practices.

The Enron emails dataset has become a popular resource for researchers studying various aspects of corporate communication, fraud detection, and natural language processing. The dataset is publicly available and has been used for a wide range of research projects, including machine learning and text mining studies. However, due to the sensitive nature of the emails, personal information and other sensitive content have been redacted to protect the privacy of individuals involved in the investigation.

Machine Learning Project Lifecycle

1. Business Problem
2. Data Acquisition
3. Data Preparation
4. EDA & Visualizations
5. Feature Engineering
6. Machine Learning Model Development
7. Machine Learning Model Evaluation
8. Business Problem:

Business problem definition is the most critical phase of a Data Science Life Cycle, if conducted well, will save lot of time, money and resources. Understand the problem by talking to the stake holders & domain experts to get the clear understanding of the problem and document all the requirements. Our business problem is to design an automated reproducible machine learning solution that is capable of classification of emails into spam or ham based on its training data.

1. Data Acquisition:

We can ask following question in data acquisition phase.

* What data do we need for our project?
* What are the data sources and data format?
* Where is the data located?
* How can we obtain the data?
* What is the most efficient way to store and access all of it for later processing?

In data acquisition, we gather or collect data from various sources that can be used for analysis, modeling, or other purposes. The process involves identifying relevant sources of data, selecting appropriate methods for collecting the data, and ensuring that the data is accurate, complete, and relevant to the task at hand.

Data acquisition can involve a variety of techniques and tools, such as web scraping, surveys, sensors, public datasets, and more. The process may also involve cleaning and preprocessing the data to ensure that it is in a format that can be easily analyzed.

Data acquisition is a critical step in many data-related tasks, including machine learning, data mining, and business intelligence. The quality and relevance of the data acquired can have a significant impact on the accuracy and effectiveness of the final analysis or model. Therefore, it is important to carefully plan and execute the data acquisition process to ensure that the resulting data is of high quality and fit for purpose.

We have the Enron emails dataset is a collection of emails and attachments from Enron Corporation. The dataset includes over 500,000 emails from approximately 150 Enron employees, including senior executives, that were collected as part of a federal investigation into the company's financial practices. In our scenario the data size is 5173 in which half are spams and half are ham emails. The data is present in well structure order. The dataset is arranged in two directories named as spam and ham. The text file names contain date of email and its label either spam or ham.

1. Data Preparation:

In data preparation, we transform and clean the acquired data into a format that is suitable for analysis. The process involves a series of steps, such as data cleaning, data transformation, feature selection, and feature engineering.

Data cleaning involves identifying and correcting errors in the data, such as missing values, inconsistent formatting, and outliers. This step is important to ensure that the data is accurate and consistent.

Data transformation involves converting the data into a format that is suitable for analysis. This may include normalizing or standardizing the data, scaling the data, or applying other mathematical transformations.

Feature selection involves identifying the most relevant features or variables in the data that are most likely to contribute to the analysis or modeling task at hand. This can help reduce the dimensionality of the data and improve the accuracy of the analysis.

Feature engineering involves creating new features or variables from the existing data that may be more informative or useful for the analysis or modeling task. This may involve combining existing features, creating new variables based on domain knowledge, or extracting features from unstructured data such as text or images.

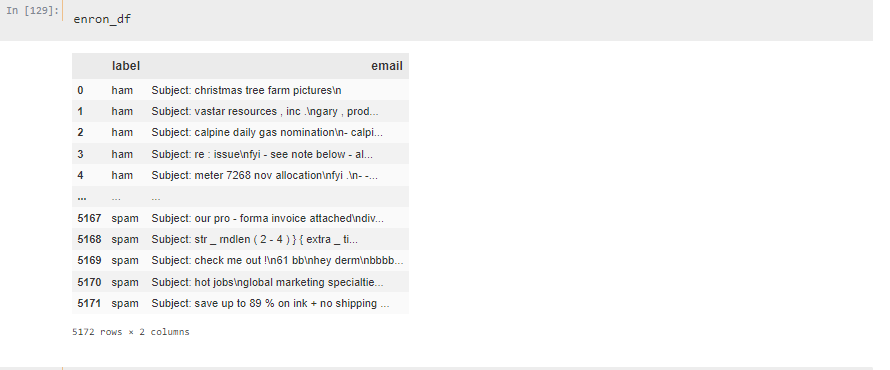
Data preparation is a critical step in many data-related tasks, and the quality of the prepared data can have a significant impact on the accuracy and effectiveness of the final analysis or model. Therefore, it is important to carefully plan and execute the data preparation process to ensure that the resulting data is of high quality and fit for purpose.

Data Reading:

As data is arranged in two directories namely spam and ham under the parent directory of enron1. The emails are in form of text files. We utilized a loop to iterate each directory and files inside that directory. The text of each file is extracted with the open() of python. Each file with its label is stored in a 'files\_dict ' dictionary. Finally, the whole dictionary is converted into a pandas data frame with column labels of 'email ' and 'label .' below figure is a demonstration of data reading.

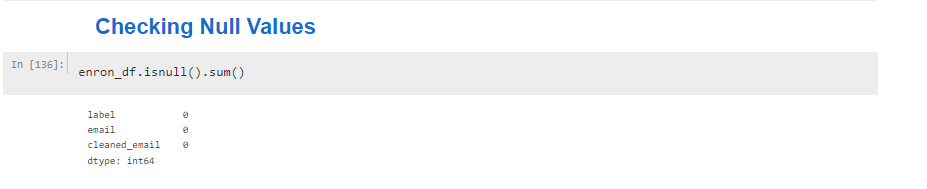


Final ouput of data reading:



Data Cleaning:

Our dataset is fully populated. There are no null values present.



In the second version of data cleaning, we removed all the text that is not important in making predictions. Text cleaning is a crucial step in natural language processing (NLP) that involves preprocessing raw text data to improve the quality of the data and make it ready for analysis. The primary objective of text cleaning is to remove any unnecessary noise or irrelevant information from the text while retaining only the relevant content.

Some of the common reasons why we need text cleaning are:

* Text normalization: Text cleaning helps in converting text data to a consistent format, making it easier to analyze. For example, converting all the text to lowercase, removing any punctuation marks or special characters, and expanding contractions like "can't" to "cannot."
* Noise reduction: Raw text data often contains a lot of irrelevant information such as HTML tags, stop words, or numbers that can interfere with text analysis. Text cleaning helps to remove such noise, leaving behind only the relevant information.
* Data standardization: Text cleaning ensures that data is standardized across all documents, making it easier to compare and analyze. For example, ensuring that dates are in a consistent format across all documents.
* Improving accuracy: Text cleaning can help improve the accuracy of machine learning models by removing any irrelevant information that may interfere with the model's performance.

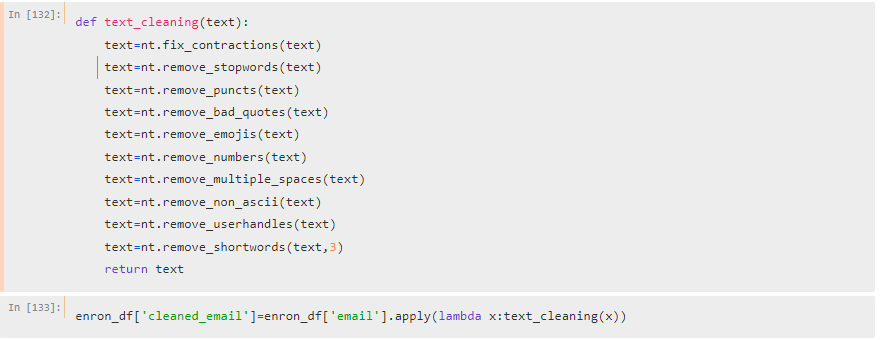
Some of the common things that are removed during text cleaning include:

* Stop words: Stop words are common words that are often removed from text data because they do not carry significant meaning. Examples of stop words include "and," "the," "a," "an," etc.
* Punctuation marks: Punctuation marks such as commas, periods, exclamation marks, and question marks are often removed from text data because they do not add much meaning.
* HTML tags: HTML tags are often present in web scraped data and need to be removed during text cleaning.
* Numbers: Numbers are often removed from text data because they do not add much meaning and can interfere with text analysis.
* Special characters: Special characters such as @, #, and $ are often removed from text data because they do not add much meaning and can interfere with text analysis.

We used neattext python package for text cleaning. The neattext provides very simple and fast way to clean the data, we just need to pass the text to function and it will return output against the specified function.

This includes:

* text=nt.fix\_contractions(text)
* text=nt.remove\_stopwords(text)
* text=nt.remove\_puncts(text)
* text=nt.remove\_bad\_quotes(text)
* text=nt.remove\_emojis(text)
* text=nt.remove\_numbers(text)
* text=nt.remove\_multiple\_spaces(text)
* text=nt.remove\_non\_ascii(text)
* text=nt.remove\_userhandles(text)
* text=nt.remove\_shortwords(text,3)



Lemmatization:

Text lemmatization is the process of reducing words to their base or root form, called a "lemma," while still maintaining the meaning of the word. The resulting lemma is a valid word that can be found in a dictionary, which makes it easier to analyze text data.

The main purpose of lemmatization in NLP is to normalize words that have the same base form, so that they can be grouped together and analyzed as a single term, instead of multiple variations of the same term. This can help in various NLP tasks such as text classification, sentiment analysis, and topic modeling.

For example, consider the following variations of the same word "run": running, ran, runs. By lemmatizing these words, we can convert them to their base form "run," which can simplify text analysis and provide better accuracy in NLP models.

To implement lemmatization in NLP, we typically use a lemmatizer, which is a software tool that applies lemmatization rules to text data. One commonly used lemmatizer in NLP is the WordNet lemmatizer, which is based on the WordNet lexical database.



1. Exploratory Data Analysis:

EDA stands for Exploratory Data Analysis, which is an approach used to analyze and summarize the main characteristics of a dataset. EDA is an essential step in data analysis and is used to gain insights into the data, identify patterns, detect anomalies and relationships, and ultimately, guide the selection of appropriate modeling techniques.

The goal of EDA is to understand the data, to identify potential issues or problems with it, and to develop hypotheses about the relationships among variables. Some of the tasks that are typically performed during EDA include:

* Data Visualization: This involves creating various visualizations such as histograms, scatter plots, and box plots to visualize the distribution of variables, identify outliers and anomalies, and detect patterns and relationships.
* Summary Statistics: This involves computing descriptive statistics such as mean, median, mode, and standard deviation to summarize the distribution of variables and to detect anomalies.
* Data Preprocessing: This involves transforming the data, scaling variables, and encoding categorical variables to prepare it for further analysis.
* Hypothesis Testing: This involves testing hypotheses about the relationships between variables, using statistical techniques such as correlation analysis, ANOVA, and t-tests.

Overall, EDA involves understanding your data and identifying patterns. It involves identifying relationships and correlations between variables using visual as well as statistical techniques. These patterns are not evident when you are looking at data in tables. A correct visualization tool can help you quickly gain a deeper understanding of your data

Word Clouds:

A word cloud is a visual representation of text data in which the size of each word represents its frequency or importance in the text. In other words, the more frequently a word appears in the text, the larger it appears in the word cloud.

Word clouds are often used to summarize and visualize the main themes or topics in a text dataset. They can provide a quick and intuitive overview of the most common words in the text, and help to identify patterns and relationships among them.

Word clouds are particularly useful for exploring text data in a qualitative and exploratory manner, and for generating insights that can guide further analysis. They are commonly used in various fields such as marketing, social media analysis, and data journalism to visualize and communicate the main ideas and concepts in a text corpus.

To create a word cloud, we typically start by preprocessing the text data to remove stop words, punctuation, and other irrelevant information. We then use a word cloud generator tool, such as Wordle or Tagxedo, to create the visualization. The resulting word cloud typically displays the most frequent words in the text dataset, with the larger words indicating higher frequency and importance.

However, it's important to note that word clouds have limitations and should be used in conjunction with other analytical techniques. Word clouds do not provide statistical analysis or insights into the relationships between words, and they can be easily biased by factors such as font size and color choices. Therefore, it's important to use word clouds in conjunction with other visualization techniques and data analysis methods.

Word cloud for the emails which are labelled as spam.



Word cloud for the emails which are labelled as ham.



NER:

NER stands for Named Entity Recognition, which is a natural language processing (NLP) technique used to identify and classify named entities in text. Named entities are words or phrases that refer to specific entities such as people, organizations, locations, dates, and other entities with specific names or labels.

SpaCy NER is a pre-built NER model available in the SpaCy library, which is a popular NLP library for Python. The SpaCy NER model uses machine learning algorithms to identify and classify named entities in text data.

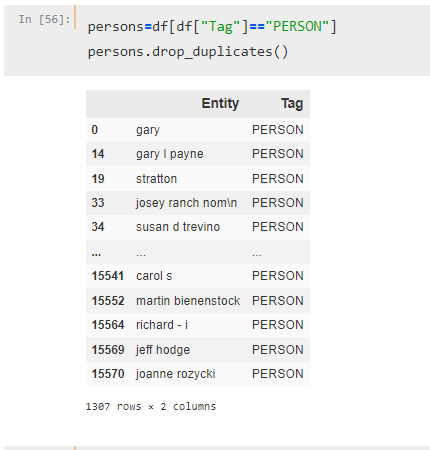
The main goal of NER is to extract structured information from unstructured text data, and to identify important entities and their relationships for further analysis. Some common use cases of NER include:

* Information Extraction: This involves extracting specific pieces of information from text data, such as names, dates, and locations, and organizing them into a structured format.
* Sentiment Analysis: This involves analyzing the sentiment or opinion expressed towards specific entities in the text data.
* Entity Linking: This involves linking named entities in text data to a knowledge graph or database to retrieve additional information about them.
* Chatbots and Virtual Assistants: NER is often used in chatbots and virtual assistants to recognize and respond to user queries more accurately.

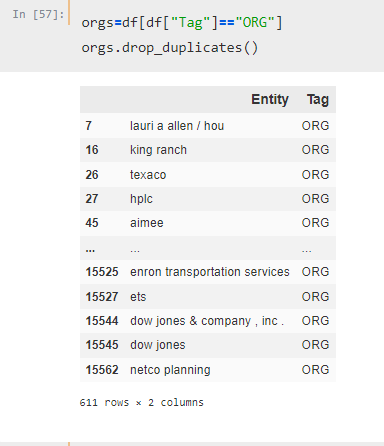
The importance of NER lies in its ability to automate the process of identifying and extracting important information from large amounts of unstructured text data. By automating this process, NER can save time and effort, and provide more accurate and consistent results compared to manual methods. This can be particularly useful in various industries such as healthcare, finance, and social media analysis, where large volumes of unstructured data need to be analyzed to extract valuable insights.

We got following results in NER ham Emails.

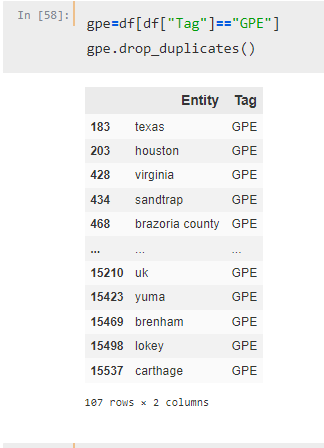
PERSONS:



Organizations:

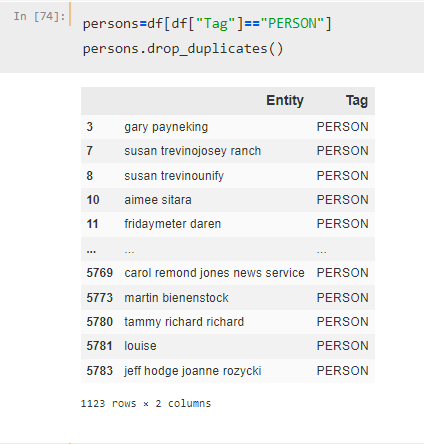


Geographic Locations:

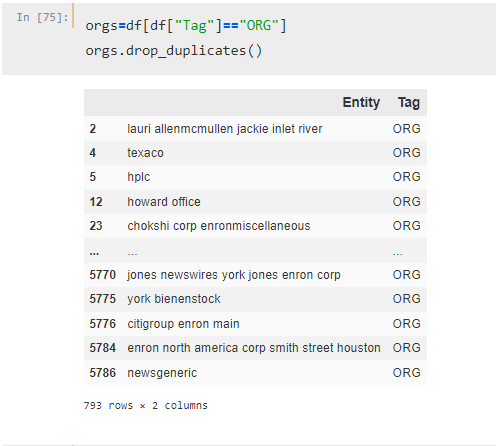


We got following results in NER spam Emails.

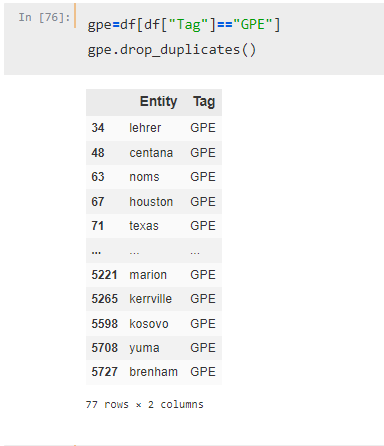
PERSONS:



Organizations:



Geographic Locations:



1. Model Development

Text Vectorization:

Text vectorization is a crucial step in natural language processing (NLP) as it converts text data into numerical vectors that machine learning models can understand and analyze. NLP models require a fixed-length input, but text data can vary in length and contain complex structures such as grammar and syntax.

Text vectorization techniques, such as bag-of-words, TF-IDF, and word embeddings, transform text into numerical representations that capture the semantic and contextual meaning of words and phrases in the text. These numerical representations enable machine learning models to make predictions, classify texts, and generate new content.

The bag-of-words technique represents text data as a collection of individual words, ignoring grammar and syntax. It creates a matrix that records the frequency of each word in the text data. The TF-IDF technique also considers the frequency of words in text data, but it assigns weights to words based on their importance in the corpus. Word embeddings, on the other hand, represent words as vectors in a high-dimensional space, where the proximity of vectors represents the semantic similarity of words.

Once text data is vectorized, it can be used as input to machine learning models, such as classification or regression models, to perform various NLP tasks, such as sentiment analysis, text classification, and machine translation. These models use the vectorized representation of text data to learn patterns and make predictions.

We used the TF-IDF as our text vectorization needs. To implement it, first, need to import the necessary libraries, including Scikit-learn (sklearn) and Pandas. Load the dataset you want to perform TF-IDF on. Create a TfidfVectorizer object and set the parameters you want to use. You can customize the parameters based on your specific use case. Apply the vectorizer to the dataset by calling the fit\_transform method of the vectorizer object. Convert the sparse matrix returned by fit\_transform into a Pandas DataFrame for easier analysis.



Train Test Split:

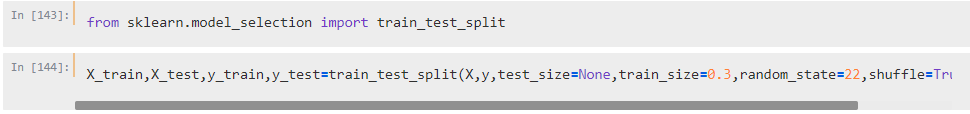
Train-test split is a technique used in machine learning to evaluate the performance of a model. It involves splitting the available dataset into two separate sets, one for training the model and the other for testing the model's performance. The training set is used to train the model, and the testing set is used to evaluate the model's performance.

The steps to implement train-test split using Scikit-learn are as follows:

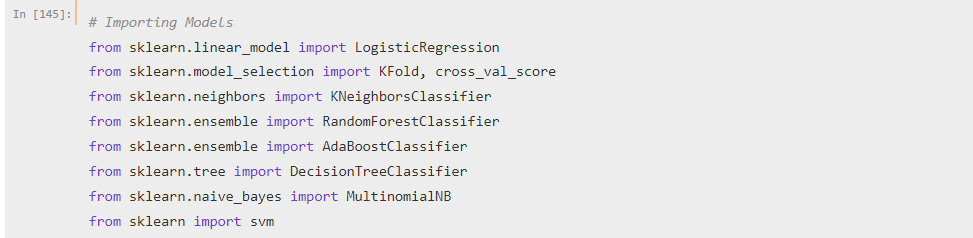
First, you need to import the necessary libraries, including Scikit-learn (sklearn) and Pandas.

Load the dataset you want to perform train-test split on.

Split the dataset into training and testing sets using the train\_test\_split function. You can specify the percentage of the data you want to allocate to the training set and testing set using the test\_size parameter. We passed test\_size=0.3 as per our requirements.



Importing Models

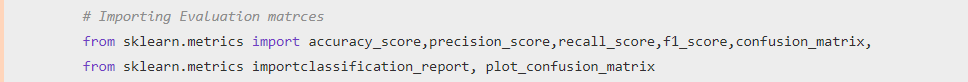


1. Model Evaluations:

Model evaluation in classification problems is the process of assessing how well a machine learning model is performing in predicting the correct class labels for a given set of input data.

There are several metrics that are commonly used for evaluating the performance of classification models:

* Accuracy: This metric measures the percentage of correctly predicted class labels out of all the predictions made by the model. While accuracy is a commonly used metric, it can be misleading in cases where the classes are imbalanced.
* Precision: This metric measures the proportion of true positives (i.e., correctly predicted positives) out of all the predicted positives. Precision is a useful metric when the focus is on minimizing false positives.
* Recall: This metric measures the proportion of true positives out of all the actual positives in the dataset. Recall is a useful metric when the focus is on minimizing false negatives.
* F1-score: This metric is the harmonic mean of precision and recall, and is often used when there is a trade-off between precision and recall.
* Confusion matrix: A confusion matrix is a table that shows the number of true positives, false positives, true negatives, and false negatives predicted by the model.



Cross Validation:

Cross-validation is a technique used in machine learning to assess the performance of a model on unseen data. It involves partitioning the available data into multiple subsets, or "folds", where each fold is used once as a validation set while the other folds are used for training.

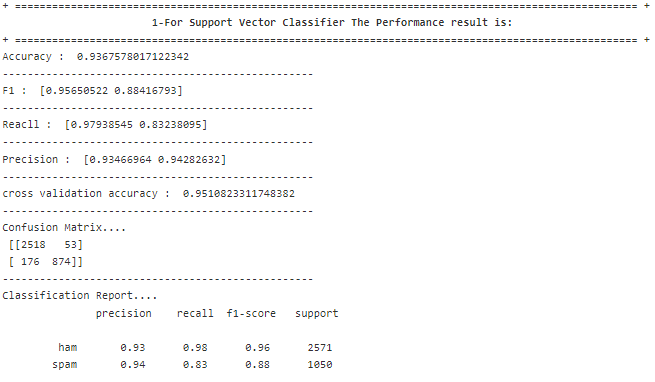
The most common type of cross-validation is k-fold cross-validation, where the data is split into k equally sized folds. The model is trained on k-1 folds, and the remaining fold is used for validation. This process is repeated k times, with each fold being used once for validation. The performance of the model is then averaged over the k folds.

Cross-validation is important because it helps to mitigate the risk of overfitting, which occurs when a model is too complex and captures the noise in the training data rather than the underlying patterns. By using multiple folds for validation, cross-validation provides a more reliable estimate of the model's performance on unseen data.

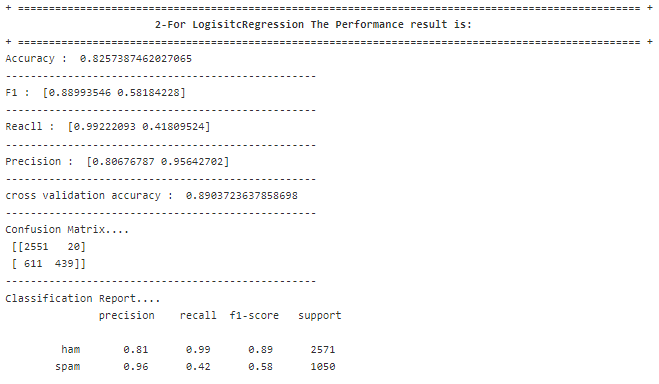
One disadvantage of cross-validation is that it can be computationally expensive, especially for large datasets or complex models. Another potential issue is that the performance of the model can vary depending on how the data is partitioned. To address this, techniques such as stratified k-fold cross-validation or leave-one-out cross-validation can be used to ensure that the data is partitioned in a representative way.



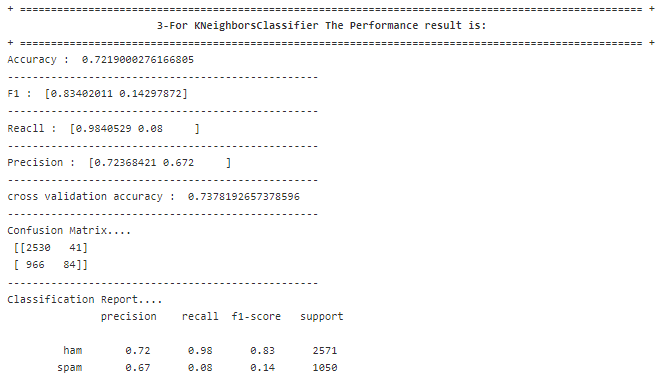
Evaluation Metrics of SVM



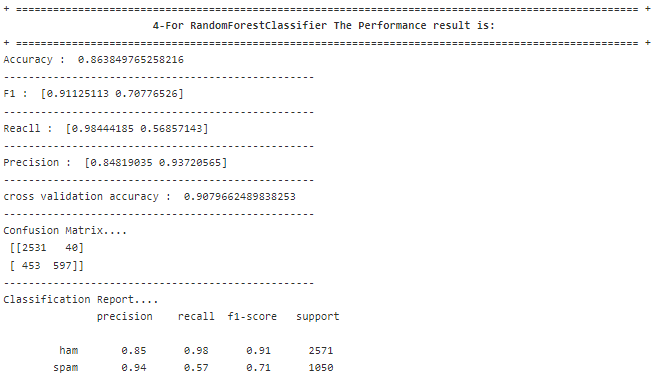
Evaluation Metrics of Logistic Regression



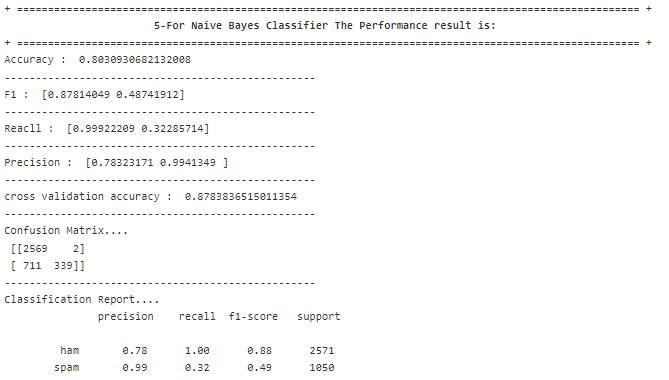
Evaluation Metrics of KNN



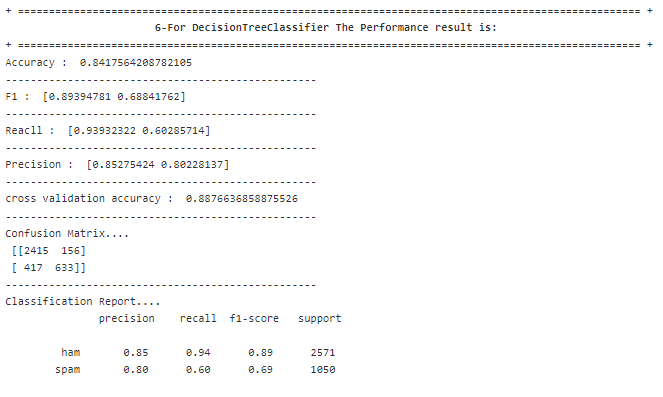
Evaluation Metrics of RFC



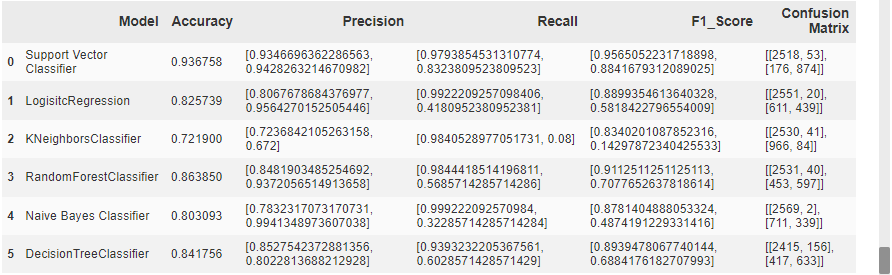
Evaluation Metrics of NB



Evaluation Metrics of DTC



Here is summary of all the models. The Support Vector Machine stands on top in terms of both F1 score and accuracy.



Final Comments:

The SVM is the best model for our data. It is taking lead in accuracy and F1 score. The 0.97 recall shows that our model has captured the data reality. Similarly, 0.95 precision represents that our model is doing a job of making predictions. The data size is not sufficient, in the future we can add more data and reproduce the results on new data.

How to Run the code:

* You should have python installed on the computer.
* You should have jupyter notebook installed on the computer.
* The data directory should be in the same place where the code file is stored.
* After these steps, just run all the cells of jupyter to reproduce the results.